



チャットに Google 認証を使用

サンプルファイル : [googleauth.html](#)

注意) Web サーバーにアップロードして確認する必要がある。
(Google 認証は HTTP 通信で利用する必要があるため)

HTML：ローディング表示箇所

id="loading_start" 内の IMG でローディングアニメーションを表示

```
7 <body>
8
9 <!-- Loading画面 -->
10 <div id="loading_start" style="height:100%;background-color:#e3f4fd;text-align:center;">
11   
12 </div>
13
14
15 <!-- コンテンツ表示画面 -->
16 <div id="loading_end" style="display:none;">
17   <button id="logout">ログアウト</button>
18   <div id="user_view">Email</div>
19   <div>名前<input type="text" id="username"></div>
20   <div>
21     <textarea rows="5" id="text"></textarea>
22     <button id="send">送信</button>
23   </div>
24   <!-- リスト表示 -->
25   <div id="output"></div>
26 </div>
27 <!-- /コンテンツ表示画面 -->
28
29
```

HTML：フォーム表示箇所

ローディング終了後、id="loading_end" を表示（ローディングは非表示に）

```
7 <body>
8
9 <!-- Loading画面 -->
10 <div id="loading_start" style="height:100%;background-color:#e3f4fd;text-align:center;">
11   
12 </div>
13
14
15 <!-- コンテンツ表示画面 -->
16 <div id="loading_end" style="display:none;">
17   <button id="logout">ログアウト</button>
18   <div id="user_view">Email</div>
19   <div>名前<input type="text" id="username"></div>
20   <div>
21     <textarea rows="5" id="text"></textarea>
22     <button id="send">送信</button>
23   </div>
24   <!-- リスト表示 -->
25   <div id="output"></div>
26 </div>
27 <!-- /コンテンツ表示画面 -->
28
29
```

1. ログイン成功時の処理を追加

```
82 //-----
83 //Googleログイン
84 //-----
85 ▼ firebase.auth().getRedirectResult().then(function(result){
86 ▼ ... if(!result.credential){
87 ... //ログインしていなければ認証画面へリダイレクト
88 ... var provider = new firebase.auth.GoogleAuthProvider();
89 ... firebase.auth().signInWithRedirect(provider);
90 ... return;
91 ... }
92 ... //ログイン成功時の処理
93 ... $("#user_view").html("Login: "+result.user.email);
94 ... $("#username").val(result.user.displayName);
95 ... $("#loading_start").fadeOut(100); //Loading終了
96 ... $("#loading_end").fadeIn(500); //Loading表示
97 ... //console.dir(result);
98 ▼ }).catch(function(error){
99 ... console.log('Error', error);
100 ... //ログイン失敗時の処理
101 ... $("#user_view").html("Login Error.");
102 ...
103 });
```

- ※ `result.user.email` で Google アカウントの Email を取得
- ※ `result.user.displayName` で Google アカウントの名前を取得

以下追加処理

```
//ログイン成功時の処理
$("#user_view").html("Login: "+result.user.email);
$("#username").val(result.user.displayName);
$("#loading_start").fadeOut(100); //ローディング終了
$("#loading_end").fadeIn(500); //ローディング表示
```

2. ログイン失敗時の処理を追加

```
82 //-----
83 //Googleログイン
84 //-----
85 ▼ firebase.auth().getRedirectResult().then(function(result){
86 ▼   ...if(!result.credential){
87     ...//ログインしていなければ認証画面へリダイレクト
88     ...var provider = new firebase.auth.GoogleAuthProvider();
89     ...firebase.auth().signInWithRedirect(provider);
90     ...return;
91   }
92   ...//ログイン成功時の処理
93   ...$("#user_view").html("Login: "+result.user.email);
94   ...$("#username").val(result.user.displayName);
95   ...$("#loading_start").fadeOut(100); //Loading終了
96   ...$("#loading_end").fadeIn(500); ...//Loading表示
97   ...//console.dir(result);
98 ▼ }).catch(function(error){
99   ...console.log('Error', error);
100   ...//ログイン失敗時の処理
101   ...$("#user_view").html("Login Error.");
102   ...
103 });
```

以下追加処理

```
-----

//ログイン失敗時の処理
$("#user_view").html("Login Error.");

-----
```

3. ログアウト処理

```
105  //-----
106  //Googleログアウト
107  //-----
108  ▼ $("#logout").on("click",function(){
109  ▼    firebase.auth().signOut().then(function(){
110    // Sign-out successful.
111    alert("Logout");
112  ▼    },function(error){
113    // An error happened.
114    });
115  });
```

以下追加処理

```
// Sign-out successful.
alert("Logout");
```

※ログアウト表示をアラートでしましょう。

4. Google 認証設定

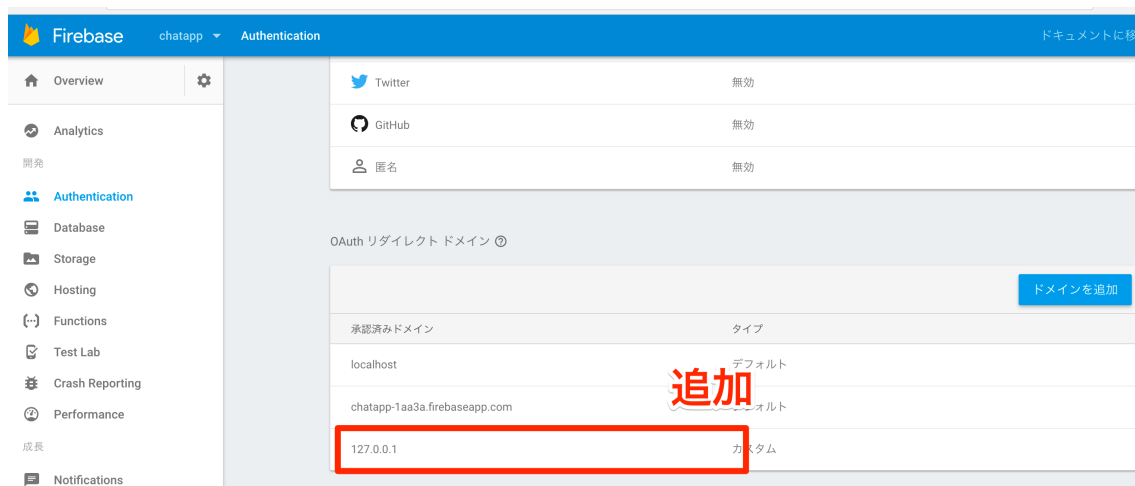
The image shows the Firebase Authentication console interface. The left sidebar contains navigation links: Overview, Analytics, Authentication (highlighted with a red box and a circled '1'), Database, Storage, Hosting, Functions, Test Lab, Crash Reporting, and Performance. The main content area is titled 'Authentication' and has tabs for 'ユーザー' (Users), 'ログイン方法' (Login Method, highlighted with a red box and a circled '2'), and 'TEMPLATES'. Below the tabs, a table lists login providers and their status:

ログインプロバイダ	ステータス
メール / パスワード	無効
電話番号	無効
Google	有効
Facebook	無効
Twitter	無効

The 'Google' row is highlighted with a red box and a circled '3'. Below this, a configuration page for Google login is shown. It includes a toggle switch for '有効にする' (Turn on), which is currently turned on. Below the toggle, there is a paragraph of text in Japanese explaining that Google login is automatically set up for iOS and web apps, but for Android, additional steps are required. There are two expandable sections: '外部プロジェクトの ID をホワイトリストに追加します (オプション)' and 'ウェブ SDK 設定 (オプション)'. At the bottom right, there are two buttons: 'キャンセル' (Cancel) and '保存' (Save, highlighted with a red box).

5. OAuth リダイレクトを設定

「127.0.0.1」を追加（Bracket エディターの場合）



※ Web サーバーにアップロードして確認する必要がある。